REMARKS

The Examiner has objected to the use of the term "servlet" in Claims 20 and 21. Applicants respectfully assert that the user of the term "servlet" is appropriate. Applicants have already recited the system server component in Claim 1, from which Claims 20 and 21 directly and indirectly depend. The additional limitation which is being recited in Claims 20 and 21 is the web servlet (or applet running on the server) which is detailed in the Specification, representatively on page 8, lines 7-11. The term "servlet" is a term of the art which is well understood by one having skill in the art. Accordingly, Applicants respectfully request reconsideration of the objection.

The Examiner has rejected Claims 1, 2, 5-7 and 9-21 as unpatentable over Warfield in view of Halviatti; and Claims 8, 22, 23 and 24 as unpatentable over Warfield in view of Brouwer. For the reasons set forth below, Applicants believe that the claims, as amended herein, are patentable over the cited prior art.

The Warfield patent describes a software testing tool which creates and tests. The Warfield system creates a set of test cases, creates populations of test scripts for the test cases, and analyzes the test using the code coverage or another fitness measure (Col. 3, lines 59-63 and Col. 4, lines 21-22). Warfield

is not selecting client machines to run tests based on the availability of the client machines and the tests; is not dynamically creating test scripts based on the tests and the machines which will run them; and is not executing the software tests on client machines and collecting the results therefrom. Warfield is simply iteratively testing and refining tests, but is not actually performing software tests on client machines.

With specific reference to the claim language, the Warfield patent does not teach or suggest accepting test requests from a user wherein each test request comprising an identifier for selecting test data from the test bucket. Warfield does not have established tests stored in a test bucket. Warfield is at a more preliminary stage wherein tests are being created and tested. With regard to the next claim feature, Warfield does not teach or suggest executing a resource process for managing system resource pool to indicate resources available for software testing on a set of client computer systems. As noted above, Warfield creates states machines to test tests, it does not deploy existing tests with dynamically created test execution script data to client machines. Next, Warfield does not teach or suggest the executing of a job execution process for dynamically creating test execution script data based on the test data identified in a test request received from the job receiver process at the job execution process and the availability of resources required for the execution of the test on one or more of the set of client

computer systems. Moreover, Warfield does not teach initiating testing at said job execution process by forwarding test execution script data to one or more of the set of client computer systems. Finally, while Warfield does analyse test results using code coverage or another fitness measure, Warfield is not accepting and storing test results from the set of client computer systems. Warfield is analyzing or testing the tests and then iteratively refining the tests to run again. Warfield does not gather test results from client machines for storage.

Applicants respectfully assert that one having skill in the art would not logically modify the teachings of the Warfield patent in such a way as to arrive at the invention as claimed. Since Warfield does not have established tests stored for use, does not have client machines from which to choose deployment, does not dynamically create test scripts based on the client machines and the established tests, it cannot be maintained that Warfield, alone or in combination with the additionally cited art, obviates the present invention.

Applicants assert that neither Halviatti nor Brouwer provide the teachings which are necessary to modify Warfield in such a way as to obviate the invention as claimed. Halviatti discloses resource management. However, there is no suggestion to combine Halviatti's dispatcher process with the internal state machine testing of Warfield. Moreover, even if one were to modify Warfield with Halviatti's resource dispatcher, one would not

arrive at the present invention since neither reference teaches or suggests storing established tests, selecting from available client machines to run tests, dynamically creating tests scripts based on the client machines and the established tests, and running the test scripts to report results.

Similarly, the Brouwer patent does not provide teachings which would logically be combined with Warfield. Brouwer discloses an application "to be able to provide a modular format to meet the need of the future system." Modifying Warfield with a modular application would not result in the claimed invention. One would simply arrive at a Warfield system which could parse ASCII formal or TCP/IP format. However, one would still not have the storing of established tests, the selection from available client machines to run tests, the dynamic creating of tests scripts based on the client machines and the established tests, and the running of test scripts to report results.

Applicants respectfully assert that the combined teachings of the cited references do not obviate the invention as claimed. Applicants have amended the claim language of the independent claims to correct an antecedent basis problem with the term "test execution script data". The amendments also consolidate the recitation of the job execution process to make the claim more readable and to highlight the distinctions over the art. No new matter has been added by the amendments.

Based on the foregoing amendments and remarks, Applicants respectfully request entry of the amendments, withdrawal of the rejections, and allowance of the claims.

Respectfully submitted,
C. Conan, et al

By: Anne Vachon Dougherty
Anne Vachon Dougherty
Attorney for Applicant
Reg. No. 30,374

MARKED UP CLAIMS WITH AMENDMENTS SHOWN

1. A system for automated testing of software, the system comprising a system server component comprising,

a test bucket for storing sets of test data,

a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

a resource process and resource pool for managing system resource data to indicate resources available for software testing on a set of client computer systems,

a job execution process for creating test [script] execution script data based on the test data identified in a test request,

wherein the job execution process [receiving] receives the test request from the job receiver process,

[the dynamic] dynamically creates the test execution script [being created] based upon the resource pool indicating the availability of resources required for the execution of the [dynamic] test [script] on one or more of the set of client computer systems, and

[the job execution process initiating] initiates testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

the system server component further comprising a means for accepting and storing test results from the set of client computer systems.

15. A computer program product for use with a computer comprising a central processing unit and random access memory,

said computer program product comprising a computer usable medium having computer readable code means embodied in said medium for software testing in distributed systems, said computer program product comprising:

computer readable program code means for causing a computer to define and manage a test bucket for storing sets of test data,

computer readable program code means for causing a computer to execute a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

computer readable program code means for causing a computer to execute a resource process for managing system a resource pool to indicate resources available for software testing on a set of client computer systems,

computer readable program code means for causing a computer to execute a job execution process for creating test [script] execution script data based on the test data identified in a test request,

wherein the job execution process [receiving] receives the test request from the job receiver process,

[the dynamic] dynamically creates the test execution script data based [being created upon] the resource pool indicating the availability of resources required for the execution of the [dynamic] test [script] on one or more of the set of client computer systems, and

[the job execution process initiating] initiates testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

computer readable program code means for causing a computer to accept and store test results from the set of client computer systems.

17. A computer program product tangibly embodying a program of instructions executable by a computer for implementing a system for automated testing of software, the system comprising a system server component comprising,

a test bucket for storing sets of test data,

a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

a resource process and resource pool for managing system resource data to indicate resources available for software testing on a set of client computer systems,

a job execution process for creating test [script] execution script data based on the test data identified in a test request, by [the job execution process] receiving the test request from the job receiver process,

dynamically creating [the dynamic] test execution script data based [being created] upon the resource pool indicating the availability of resources required for the execution of the [dynamic] test [script] on one or more of the set of client computer systems, and

[the job execution process] initiating testing by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

the system server component further comprising a means for accepting and storing test results from the set of client computer systems.

19. A method for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium for software testing

in distributed systems, said method comprising the steps at said computer of:

defining and managing a test bucket for storing sets of test data,

executing a job receiver process, for accepting test requests from a user, each test request comprising an identifier for selecting test data from the test bucket,

executing a resource process for managing system a resource pool to indicate resources available for software testing on a set of client computer systems,

executing a job execution process for creating test [script] execution script data based on the test data identified in a test request, by performing the steps of:

receiving the test request from the job receiver process at the job execution process,

dynamically creating a [dynamic] test execution script indicating the availability of resources required for the execution of the [dynamic] test [script] on one or more of the set of client computer systems, and

initiating testing at said job execution process by forwarding the test execution script data to the appropriate one or more of the set of client computer systems, and

accepting and storing test results from the set of client computer systems.